Homework 3 Solutions

1. Prove by induction that if the annihilator for a recurrence is $(E-b)^n$, for any positive integer, n, and constant, b, the solution to the recurrence is

$$a_i = b^i \left(\sum_{j=1}^{n} c_j i^{j-1} \right), \forall i \in N$$

in which the c_j are constants to be determined from initial conditions.

Solution: (a) For the base case, it is shown that the summation is the solution of the recurrence by applying the annihilator $(E-b)^n$ with $n=1$ to yield zero.

$$\begin{aligned} (E-b) < c_j b^i > &= E < c_j b^i > -b < c_j b^i > \\ &= < c_j b^{i+1} > - < c_j b^{i+1} > \\ &= < 0 > \end{aligned}$$

(b) Now assuming that $(E-b)^n$ is the annihilator for the recurrence for any positive integer n, then

$$(E-b)^n < \left(\sum_{j=1}^{n} c_j i^{j-1} \right) b^i > = < 0 > \qquad (1)$$

Would it still be the annihilator in the inductive step $n+1$, i.e.

$$(E-b)^{n+1} < \left(\sum_{j=1}^{n+1} c_j i^{j-1} \right) b^i > = < 0 > ?$$

To show this we first multiply the new summation with $(E-b)^n$

$$\begin{aligned} (E-b)^n < \left(\sum_{j=1}^{n+1} c_j i^{j-1} \right) b^i > &= (E-b)^n < \left(\sum_{j=1}^{n} c_j i^{j-1} + c_{n+1} i^n \right) b^i > \\ &= (E-b)^n < \left(\sum_{j=1}^{n} c_j i^{j-1} \right) b^i > + (E-b)^n < c_{n+1} i^n b^i > \end{aligned}$$

First part is identical to equation (1) above, hence

$$< 0 > + (E-b)^n < c_{n+1} i^n b^i > = c_{n+1} < ((E-b)(i b^{\frac{i}{n}}))^n >$$

Let $k = i/n$, then

$$c_{n+1} < ((E-b)(nkb^k))^n >$$

Since n is a constant, it can be factored out

$$\begin{aligned} c_{n+1} n^n < ((E-b)(kb^k))^n > &= c_{n+1} n^n < (E(kb^k) - b(kb^k))^n > \\ &= c_{n+1} n^n < ((k+1)b^{k+1}) - (kb^{k+1}))^n > \\ &= c_{n+1} n^n < (b^{k+1}(k+1-k))^n > \\ &= < c_{n+1} n^n b^{(k+1)n} > \end{aligned}$$

Substituting $k = i/n$ yields

$$< c_{n+1} n^n b^{i+n} >$$

It is now straight forward to show the inductive step $n + 1$

$$(E - b)^{n+1} < \left(\sum_{j=1}^{n+1} c_j i^{j-1} \right) b^i >= (E - b)(E - b)^n < \left(\sum_{j=1}^{n+1} c_j i^{j-1} \right) b^i >= (E - b) < c_{n+1} n^n b^{i+n} >$$

This is like the equation in the base case, thus similarly

$$(E - b) < c_{n+1} n^n b^{i+n} >=< 0 >$$

This completes the proof by induction.

■

2. Consider the simple recurrence $a_i = -a_{i-2}$ with specified initial conditions a_0 and a_1. By inspecting the recurrence, write down an simple expression for a_n in terms of a_0 and a_1 (the expression may contain "if" conditions). Using annihilators, solve the recurrence, and show that it yields an expression equivalent to your first one.

Solution: By simply writing down the terms note that the sequence looks like this

$$a_n = \{a_0, a_1, -a_0, -a_1, a_0, a_1, -a_0, -a_1 ...\}$$

which can be rewritten as following

$a_n = \{a_0$, if n mod4 $= 0$

a_1, if n $mod4 = 1$

$-a_0$, if n $mod4 = 2$

$-a_1$, if n $mod4 = 3\}$

Now using anihilators note that

$E^2 \langle a_0, a_1, -a_0, -a_1, a_0, a_1, -a_0, -a_1 ...\rangle = \langle -a_0, -a_1, a_0, a_1, -a_0, -a_1 ...\rangle$

Thus $E^2 \langle ...\rangle + 1\langle ...\rangle = 0$

So the anihilator is $(E^2 + 1)$, which is $(E - (-i))(E - i) = 0$

The recurrence is therefore $c_1(-i)^n + c_2(i)^n$ which can be rewritten as

$a_n = \{c_1 + c_2$, if n mod4 $= 0$

$-c_1 i + c_2 i$, if n $mod4 = 1$

$-c_1 - c_2$, if n $mod4 = 2$

$c_1 i - c_2 i$, if n $mod4 = 3\}$

From the initial conditions we have

$a_0 = c_1 + c_2$

$a_1 = -c_1 i + c_2 i$, which reduces to the original equation for a_n

■

3. Solve the following recurrence: $a_i = 4a_{i-1} - 4a_{i-2} + i2^i + 2$ for the initial conditions $a_0 = 2$, $a_1 = 10$.

   Solution: The homogeneous annihilator for $a_i = 4a_{i-1} - 4a_{i-2} = 0$ is $(E - 4E + 4) = (E-2)^2$ and that for $i2^i + 2$ is $(E-2)^2(E-1)$. The complete annihilator is thus $(E-2)^4(E-1)$. Since in general, $(E-a)^n$ annihilates $p(i)a^i$ where $p(i)$ is any polynomial in i of degree $n-1$. We have

   $$a_i = C_1 2^i + C_2 i 2^i + C_3 i^2 2^i + C_4 i^3 2^i + C_5$$

   We are left to calculate the constants C_i. Using the initial conditions and the recurrence, we have $a_0 = 2, a_1 = 10, a_2 = 42, a_3 = 154, a_4 = 514$. Now plugging each of these numbers into the left side of the above equation and solve the resulting linear equation systems, we get $C_1 = 0, C_2 = 10/3, C_3 = 1/2, C_4 = 1/6, C_5 = 2$. Hence our solution is

   $$a_i = \frac{10}{3} i 2^i + \frac{1}{2} i^2 2^i + \frac{1}{6} i^3 2^i + 2$$

   ■

4. Consider the following family of recurrences:

   $$\sum_{k=0}^{n} \binom{n}{k} a_{i-k} = 0.$$

   Determine the general form of the solution to these recurrences for any integer $n \geq 1$. Note that no initial conditions are given; therefore, you may leave constants in the solution. Prove the correctness of your approach.

   Solution: The characteristic equation of this recurrence relation is

   $$\sum_{k=0}^{n} \binom{n}{k} r^{n-k} = 0.$$

   since

   $$\sum_{k=0}^{n} \binom{n}{k} r^{n-k} = (r+1)^n.$$

   There is a single root r = -1 of the characteristic equation. By Theorem 4 (ROSEN P325) the solution of this recurrence relation is of the form

   $$a_i = (\alpha_0 + \alpha_1 i + \alpha_2 i^2 + ... + \alpha_{n-1} i^{n-1})(-1)^i$$

   ■

[In all the following problems, for divide-and-conquer recurrences, full credit will be given for tight asymptotic bounds (including your reasoning). Also, partial credit may be given if you are only able to obtain distinct upper and lower bounds]

5. **Solve the following recurrence**

   $$T(n) = 5T(n/4) + n \log \log n$$

   Solution: $\boxed{T(n) = \Theta(n^{log_4 5})}$=$\Theta(n^{1.16})$ by the Master Theorem.

   ■

6. **Solve the following recurrence**

   $$T(n) = \log n + 2\sqrt{n} \cdot T(\lfloor \sqrt{n} \rfloor)$$

Solution: $\boxed{T(n) = \Theta(n \log n)}$

First,we apply domain transformation. Let $n = 2^m (m = \log n)$ and $S(m) = T(2^m)$

$$\begin{aligned} S(m) = T(2^m) &= T(n) \\ &= \log n + 2\sqrt{n} \cdot T(\lfloor \sqrt{n} \rfloor) \\ &= m + 2 \cdot 2^{m/2} \cdot T(2^{m/2}) \\ &= m + 2 \cdot 2^{m/2} \cdot S(m/2) \\ 2^{-m} \cdot S(m) &= m \cdot 2^{-m} + 2 \cdot 2^{-m/2} \cdot S(m/2) \end{aligned}$$

Then we apply range transformation.Let $R(m) = 2^{-m} \cdot S(m)$

$$\begin{aligned} R(m) = 2^{-m} \cdot S(m) &= m \cdot 2^{-m} + 2 \cdot 2^{-m/2} \cdot S(m/2) \\ &= m \cdot 2^{-m} + 2 \cdot R(m/2) \\ &= 2 \cdot R(m/2) + m \cdot 2^{-m} \end{aligned}$$

According to the master method, $R(m) = \Theta(m)$.Undoing range transformation, $S(m) = 2^m \cdot R(m) = \Theta(m \cdot 2^m)$.Undoing domain transformation, $T(n) = T(2^m) = S(m) = \Theta(m \cdot 2^m) = \Theta(n \log n)$. ■

7. Solve the following recurrence

$$T(n) = T(\lfloor n - \log n \rfloor) + 1$$

Solution: $\boxed{T(n) = \Theta(n/\log n)}$

By expanding the recurrence, it's clear that $T(n)$ is just the recursion depth. Observe that $\log n$ is just the number of binary bits needed in the representation of n. By expanding the recurrence, it's easy to see that at every level of the expansion, we simply take the quantity of the last level, let it being n^*, subtract $\log n^*$ from it and set the result as the new n^* to pass onto next level.

(a) We first give an upper bound on the recurrence depth. We partition the range $[1...n]$ into intervals $I_i = [2^i, ..., 2^{i+1}]$, for $i = 0, \ldots, \lg n$. The number of binary bits in the representation of a number n^* remains the same so long as n^* falls in interval $I_{\log n^*}$. Initially, $n^* = n$ is somewhere in interval $I_{\log n}$. Everytime we subtract $\log n^*$ from n^* and it may or may not drop to the next interval. Anyway, we can bound the total steps during the entire process until n^* drops to equal an initial condition value such as 1. Since there are exactly 2^i distinct values in interval I_i and so long as n^* is in I_i, we always subtract i from it, the contribution of each interval I_i to the total steps is at most $\frac{2^i}{i}$. $T(n) = O(\sum_{i=1}^{(\log n)} 2^i/i) = O(n/\log n)$.

(b) As for the lower bound, observe that as we go down the recursion tree, the term $\log n^*$ is decreasing. More steps are needed to get to the initial condition in $T(n)$ than in the case if we subtract a fixed $\log n$ every time. Therefore $T(n) \geq n/\log n = \Omega(n/\log n)$. Thus, $T(n) = \Theta(n/\log n)$

■

8. Solve the following recurrence

$$T(n) = T(\lfloor n/\log n \rfloor) + \log n$$

Solution: $\boxed{T(n) = \Theta((\log n)^2/\log\log n).}$

You must note that you can **not** apply the Master theorem here, as $\log n$ is not a constant.

First stage, we have to bound the depth of the recursion tree. The recursion depth can be written as the following recurrence $T^*(n) = T^*(n/\log n) + 1$. Clearly, $n/\log n$ has $\Theta(\log(n) - \log\log n)$ bits. Namely, we have to apply the recurrence $\frac{1}{2}(\log n/\log\log n) = \Theta(\log n/\log\log n)$ times before the number of digits representing n drops by a factor of two.

Formally, we partition the range $[1...n]$ into intervals $I_i = [2^{2^i}, ..., 2^{2^{i+1}}]$, for $i = 0, \ldots, \lg\lg n$. By the above argument, the contribution of interval I_i to $T^*(n)$ is $\Theta(2^i/i)$. Thus,

$$T^*(n) = \Theta\left(\sum_{i=0}^{\log\log n} \frac{2^i}{i}\right).$$

And it turns out that $T(n)$ is just the sum of the numbers of the digits of n at each level of recursion.

$$T(n) = \Theta\left(\sum_{i=0}^{\log\log n} \frac{2^i}{i} 2^i\right).$$

This summation can be solved exactly using known techniques. However, it is easier here to realize that for i large enough, the elements of this summation behave like a increasing geometric series, and as such the last element is proportional to the sum of the series. Thus, $T(n) = \Theta((\log n)^2/\log\log n)$.

■